

CLAIMS:

1 1. A method for recovering a global history vector in the event of a non-branch
2 flush comprising the steps of:

3 receiving a tag, wherein said tag is associated with a flush operation;

4 resetting a next-to-write pointer in a queue using said received tag;

5 reading a global history vector from said queue in an entry indexed by said
6 tag;

7 inserting a copy of said global history vector read into a global history vector
8 mechanism, wherein said global history vector mechanism is configured to update
9 said global history vector in a first mode; and

10 transmitting a command to said global history vector mechanism to enter a
11 second mode if said flush operation is a flush to a group of instructions that contains
12 no branch instructions and said tag does not equal said next-to-write pointer, wherein
13 said global history vector mechanism does not update said global history vector in
14 said second mode.

1 2. The method as recited in claim 1, wherein said command is transmitted to said
2 global history vector mechanism to enter said second mode if said tag points to an
3 entry in said queue containing information regarding a branch instruction previously
4 fetched between an address flushed to and a last address fetched at a time of said
5 flush operation.

1 3. The method as recited in claim 1, wherein if said flush operation is a flush to a
2 group of instructions that contains no branch instructions and said tag does not equal
3 said next-to-write pointer then the method further comprises the steps of:

4 scanning a copy of a selected instruction; and

5 determining if said selected instruction is a branch instruction;

6 wherein if said first instruction is not said branch instruction, then the method
7 further comprises the step of:

8 scanning a copy of a next instruction.

1 4. The method as recited in claim 3, wherein if said selected instruction is a
2 branch instruction, then the method further comprises the step of:

3 transmitting a command to said global history vector mechanism to enter said
4 first mode of operation.

1 5. The method as recited in claim 4, wherein if said first instruction is said
2 branch instruction, then the method further comprises the step of:

3 updating said global history vector upon each successive fetch of a group of
4 instructions in said first mode.

1 6. The method as recited in claim 1, wherein if said flush operation is a flush to a
2 group of instructions that contains a branch instruction then the method further
3 comprises the step of:

4 transmitting a command to said global history vector mechanism to continue
5 operation in said first mode.

1 7. The method as recited in claim 1, wherein if said flush operation is a flush to
2 a group of instructions that contains no branch instructions and said tag does not
3 equal said next-to-write pointer then the method further comprises the steps of:

4 setting a counter to a predetermined value;

5 transmitting an instruction; and

6 decrementing said counter if said instruction transmitted is not a branch
7 instruction.

1 8. The method as recited in claim 7, wherein if said flush operation is a flush to a
2 group of instructions that contains no branch instructions and said tag does not equal
3 said next-to-write pointer then the method further comprises the step of:

4 determining if said counter equals zero, wherein if said counter equals zero
5 then the method further comprises the step of:

6 transmitting a command to said global vector history mechanism to
7 reconstruct said global history vector.

1 9. The method as recited in claim 8, wherein if said counter equals zero then the
2 method further comprises the step of:

3 reconstructing said global history vector by appending a number of zero's in
4 said global history vector, wherein said number equals said predetermined value.

1 10. The method as recited in claim 9, wherein if said counter equals zero then the
2 method further comprises the step of:

3 continuing reconstructing said global history vector by updating said global
4 history vector upon each successive fetch of a group of instructions.

1 11. A system, comprising:

2 an instruction fetch unit configured to fetch instructions, wherein said
3 instruction fetch unit comprises:

4 a branch prediction logic unit, wherein said branch prediction logic
5 unit is configured to predict if a branch instruction will be taken or not taken, wherein
6 said branch prediction logic unit comprises:

7 a group history vector mechanism configured to manage a
8 group history vector; and

9 a queue coupled to said branch prediction logic unit, wherein
10 said queue is configured to store information about branch instructions, wherein said
11 queue comprises:

12 logic for receiving a tag, wherein said tag is associated
13 with a flush operation;

14 logic for resetting a next-to-write pointer in said queue
15 using said received tag;

16 logic for reading said global history vector from said
17 queue in an entry indexed by said tag; and

18 logic for transmitting a command to said global history
19 vector mechanism to enter a second mode if said flush operation is a flush to a group
20 of instructions that contains no branch instructions and said tag does not equal said
21 next-to-write pointer, wherein said global history vector mechanism does not update
22 said global history vector in said second mode.

1 12. The system as recited in claim 11, wherein said command is transmitted to
2 said global history vector mechanism to enter said second mode if said tag points to
3 an entry in said queue containing information regarding a branch instruction
4 previously fetched between an address flushed to and a last address fetched at a time
5 of said flush operation.

1 13. The system as recited in claim 11, wherein said instruction fetch unit further
2 comprises:

3 a branch scan logic unit coupled to said branch prediction logic unit, wherein
4 said branch scan logic unit comprises:

5 logic for scanning a copy of a selected instruction if said flush
6 operation is a flush to a group of instructions that contains no branch instructions and
7 said tag does not equal said next-to-write pointer;

8 logic for determining if said selected instruction is a branch
9 instruction; and

10 logic for scanning a next instruction if said first instruction is not said
11 branch instruction.

1 14. The system as recited in claim 13, wherein said branch scan logic unit further
2 comprises:

3 logic for transmitting a command to said global history vector mechanism to
4 enter said first mode if said first instruction is said branch instruction.

1 15. The system as recited in claim 14, wherein said global history vector
2 mechanism comprises:

3 logic for updating said global history vector upon each successive fetch of a
4 group of instructions in said first mode.

1 16. The system as recited in claim 11, wherein said queue further comprises:

2 logic for transmitting a command to said global history vector mechanism to
3 continue operation in said first mode if said flush operation is a flush to a group of
4 instructions that contains a branch instruction.

1 17. The system as recited in claim 11, wherein said global history vector
2 mechanism comprises:

3 logic for setting a counter to a predetermined value if said flush operation is a
4 flush to a group of instructions that contains no branch instructions and said tag does
5 not equal said next-to-write pointer;

6 wherein said instruction fetch unit further comprises:

7 a cache memory, wherein said cache memory comprises:

8 logic for transmitting an instruction if said flush operation is a flush to
9 a group of instructions that contains no branch instructions and said tag does not
10 equal said next-to-write pointer; and

11 wherein said global history vector mechanism further comprises:

12 logic for decrementing said counter if said instruction
13 transmitted is not a branch instruction.

1 18. The system as recited in claim 17, wherein said instruction fetch unit further
2 comprises:

3 a branch scan logic unit coupled to said cache memory, wherein said branch
4 scan logic unit comprises:

5 logic for determining if said counter equals zero if said flush operation
6 is a flush to a group of instructions that contains no branch instructions and said tag
7 does not equal said next-to-write pointer; and

8 logic for transmitting a command to said global vector history
9 mechanism to reconstruct said global history vector if said counter equals zero.

1 19. The system as recited in claim 18, wherein said group history vector
2 mechanism further comprises:

3 logic for reconstructing said global history vector by appending a number of
4 zero's in said global history vector if said counter equals zero, wherein said number
5 equals said predetermined value.

1 20. The system as recited in claim 19, wherein said group history vector
2 mechanism further comprises:

3 logic for continuing reconstructing said global history vector by updating said
4 global history vector upon each successive fetch of a group of instructions if said
5 counter equals zero.